



A Novel Lightweight Defense Method Against Adversarial Patches-Based Attacks on Automated Vehicle Make and Model Recognition Systems

Abdul Jabbar Siddiqui¹ · Azzedine Boukerche¹

Received: 23 September 2020 / Revised: 1 April 2021 / Accepted: 27 April 2021 /
Published online: 31 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In smart cities, connected and automated surveillance systems play an essential role in ensuring safety and security of life, property, critical infrastructures and cyber-physical systems. The recent trend of such surveillance systems has been to embrace the use of advanced deep learning models such as convolutional neural networks for the task of detection, monitoring or tracking. In this paper, we focus on the security of an automated surveillance system that is responsible for vehicle make and model recognition (VMMR). We introduce an adversarial attack against such VMMR systems through adversarially learnt patches. We demonstrate the effectiveness of the developed adversarial patches against VMMR through experimental evaluations on a real-world vehicle surveillance dataset. The developed adversarial patches achieve reductions of up to 48% in VMMR recall scores. In addition, we propose a lightweight defense method called *SIHFR* (stands for *Symmetric Image-Half Flip and Replace*) to eliminate the effect of adversarial patches on VMMR performance. Through experimental evaluations, we investigate the robustness of the proposed defense method under varying patch placement strategies and patch sizes. The proposed defense method adds a minimal overhead of less than 2ms per image (on average) and succeeds in enhancing VMMR performance by up to 69.28%. It is hoped that this work shall guide future studies to develop smart city VMMR surveillance systems that are robust to cyber-physical attacks based on adversarially learnt patches.

Keywords Cyber-physical systems security · Adversarial attacks · Vehicle recognition · Adversarial patches · Adversarial robustness

✉ Abdul Jabbar Siddiqui
aj.siddiqui@uottawa.ca

Extended author information available on the last page of the article

1 Introduction

The surveillance systems such as automated Vehicle Make and Model Recognition (VMMR) systems are essential components in smart cities and intelligent transportation systems to aid in ensuring safety and security of life, property, critical infrastructures and in security management of cyber-physical systems. The state-of-the-art VMMR systems are based on advanced deep learning models such as convolutional neural networks (CNNs). These are highly vulnerable to a new kind of cyber-physical attack that leverages adversarial machine learning. Recent studies have shown that the CNNs could be tricked or fooled to evade detection or cause mis-classification [1–4]. One of the ways in which this is achieved involves crafting or modifying the inputs through adversarially learnt patterns printed or patched on the objects, presenting a unique kind of cyber-physical attack. For example, the work of [5] developed adversarial posters and stickers to cause object detectors to not detect stop signs which is a potentially lethal attack against connected and autonomous vehicles. As another example, CNNs trained to detect and recognize persons were not able to detect them due to the placement of adversarial patches on those persons [2].

To the best of our knowledge, no prior studies have investigated the adversarial robustness of CNNs-based VMMR systems against such attacks. In this paper, we study adversarial patches-based attacks specifically targeted against VMMR systems (see Fig. 1), a problem that has not been addressed in the literature to the best of our knowledge. Moreover, we propose a lightweight defense to mitigate the effect of adversarial patches, leveraging symmetry in vehicles' frontal (or rear) faces. The proposed defense method does not require additional hardware. It can be deployed practically as a complementary component working in tandem with, or incorporated into, an automated VMMR software. Possible adopters of this technology may include smart cities (for automated surveillance), security agencies and traffic analysts.

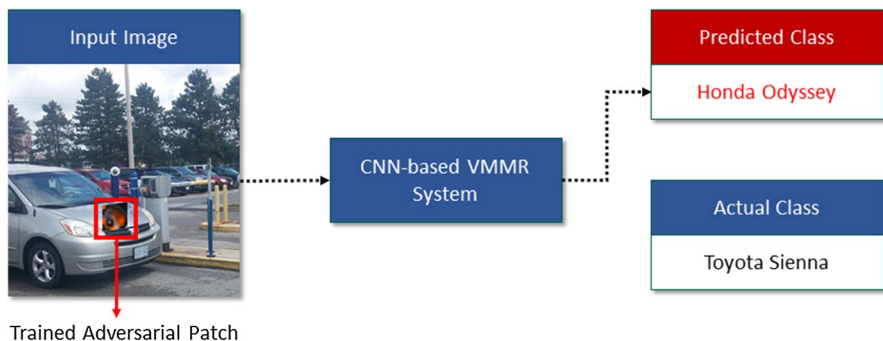


Fig. 1 (Color online) Illustration of a cyber-physical attack in action: using an adversarial patch placed on a vehicle to fool a VMMR surveillance system that is based on deep learning models such as CNNs. In this paper, we show how such adversarial patches could trick the VMMR system to mis-classify vehicles to wrong classes in order to evade detection or impersonate as another vehicle. In addition, we propose a lightweight defense method to mitigate such attacks

In areas where security is an important concern, automated VMMR systems find their applications in recognizing the makes and models of vehicles from the images captured by surveillance cameras [6, 7]. Examples of such areas include parking lots of airports, malls, checkpoints, critical facilities, etc. Moreover, automated VMMR systems serve as efficient and fast solutions to hunt suspects or targets, e.g., in aiding police to search for a vehicle of certain make and model. Malicious entities could leverage adversarial patches to attack and circumvent VMMR systems to gain unauthorized access or avoid being found. There could be two broad types of possible attacks: (i) impersonation and (ii) dodging. In the former, an adversary's goal is to impersonate a particular vehicle make and model by tricking the VMMR system. In the latter, an adversary's goal is to evade correct recognition of its make and/or model by tricking the VMMR system to recognize it as any other make and/or model.

The specific research questions that motivate this study include: (i) How feasible is it to launch adversarial attacks against CNN-based VMMR systems, (ii) How successful could adversarial attacks be in tricking a VMMR system to mis-classify vehicles (in terms of impact on precision and recall scores), and (iii) Could we develop a lightweight defense method, without requiring to modify or re-train the CNN model, to eliminate or reduce the influence of adversarial patches? Based on these, the main contributions this paper makes are as follows:

- Study the learning of adversarial patches to fool CNN-based VMMR systems,
- Investigate the impact of adversarial attacks based on these learnt patches in reducing precision and recall scores of a CNN-based VMMR system,
- Design, develop and evaluate a lightweight defense method to improve VMMR system's performance under adversarial patches-based attacks,
- Investigate the robustness of the proposed defense method under varying patch placement strategies and patch sizes, and
- Demonstrate how the proposed defense method outperforms a state-of-the-art defense method.

The remainder of the paper is organized as follows. Section 2 provides an overview of the background and related works. Section 3 describes the adversarial patches-based attacks against VMMR systems. Section 4 presents the proposed lightweight defense method. Section 5 describes the experimental setup, dataset, and the training of adversarial patches. Section 6 provides the results and discussions on the evaluation of adversarial patches-based attacks on VMMR, impact of patch placement location, and the effectiveness of the proposed defense method. The paper finally concludes in Sect. 7.

2 Background and Related Work

In this section, we first provide an overview of adversarial patches-based attacks on CNN-based object detection and image classification models. Then, we briefly describe and highlight the limitations of recent relevant related works that have proposed defense methods to mitigate the effect of adversarial patches.

2.1 Adversarial Patches-Based Attacks on Object Detection and Image Classification Models

Very recently, the problem of adversarial attacks against deep learning models has gained attention. Although several studies have been made on adversarial attacks against general object detectors, to the best of our knowledge, our work is first to investigate adversarial attacks against VMMR systems using adversarially learnt patches. In this section, we present some of the recent works on adversarial patches-based attacks against object detection and image classification CNNs.

Adversarial patches are used to perform two broad types of attacks against object detectors and image classifiers, namely: *targeted* and *non-targeted*. While the former seeks to cause the classifier to output a certain target class regardless of what object is in the scene, the latter seeks to cause the classifier to miss-classify objects of all classes to any other class. The *targeted* adversarial attacks could be used to conduct impersonation attacks, i.e., pretending or appearing as the target object or class. The *non-targeted* adversarial attacks on the other hand could be used to cause dodging attacks, i.e., the adversary just intends to avoid being detected or classified as its true self [8].

The pioneering work of [9] developed targeted adversarial attacks by learning patches that cause the classifier to produce a specific target class as output. These patches were applied in real physical scenes with great effectiveness. The patch training process involves optimizing (minimizing) the target class' expected probability. The work of [2] studied adversarial patches targeted against person detection. It was demonstrated that adversarially learnt patches could fool a popular object detector such as YOLO2 [10], causing it to miss detecting persons in the input images.

Adversarial patches that could perform both targeted and non-targeted attacks were proposed in DPATCH [11]. To learn patches for non-targeted adversarial attacks, DPATCH seeks to maximize the object detector's loss with respect to the ground truth class labels and bounding box parameters. In their work, the authors demonstrated how DPATCH trained to attack one CNN model (e.g., Faster RCNN [12]) could effectively attack another model (e.g. YOLO).

The authors of [8] proposed learning adversarial regions shaped as eye-glass frames to attack face recognition models. The adversarial regions are placed on the face as printed eye-glasses and are trained to conduct targeted and non-targeted attacks. With targeted adversarial printed eye-glasses frames, an adversary could perform an impersonation attack. On the other hand, with non-targeted adversarial printed eye-glasses frames, an adversary could perform a dodging attack (i.e., to be mis-identified as any other face).

Adversarial posters and stickers were proposed by [5] to cause object detectors to not detect stop signs which is a potentially lethal attack against connected and autonomous vehicles. Moreover, their work trained adversarial stickers that were placed on flat objects (not stop signs) and caused object detectors to mis-detect these as stop signs.

The works mentioned above study adversarial attacks against deep neural networks that are purposed for object detection, image classification or focused on

specific applications such as face recognition, person detection, stop sign detection, etc. The problem of VMMR is different from these application domains and poses a peculiar set of challenges. These challenges include multiplicity (variety of appearances of a single make-model class) and inter-class or intra-class ambiguities (similarities in appearances of different make-model classes or of different models within the same manufacturer class, respectively) [6].

On the other hand, in the case of stop signs for example, these commonly have the same shape and color. In the case of person detection, though the appearance of “person” objects varies (e.g. due to size, clothes or skin), there is a general outline or figure of persons. In the case of face recognition, multiplicity issues may occur due to factors like aging or facial hair, etc. Moreover, there is a wider area of placing adversarial patches on vehicles in comparison to other objects such as stop signs or persons.

Hence, to learn adversarial attacks against deep learning models that have been trained to overcome the VMMR challenges becomes more complicated. To the best of our knowledge, no prior study has investigated adversarial patches-based attacks against VMMR systems. We believe this work shall pave the way forward for future studies in developing adversarially robust VMMR and surveillance systems for secure smart cities.

2.2 Defense Methods Against Adversarial Patches-Based Attacks

In the literature, only a few defenses could be found, to date, against physical world adversarial patches-based attacks that target CNNs. We review state-of-the-art defenses against such patch-based adversarial attacks and qualitatively discuss the limitations of these works in comparison to our method.

The work of [1] recently proposed a defense method that involves extracting ally patches from input images through counter-processing them based on their intrinsic information contents. In their method, from each input image, a set of patches are extracted to feed the object detector. These set of patches may include patches which enclose the adversarial patch fully or partially. While an adversarial patch partially enclosed in an ally patch is most likely to be ineffective in tricking the detector, a fully enclosed one may lead to a wrong output from the detector for that particular ally patch. However, since the final classification output is based on the predictions from each patch in the alliance, the effect of the adversarial patch may most likely be eliminated. A major limitation in this method is that the object detector network has to be executed multiple times per input image. Moreover, this method would require the object detector or classifier network to be trained to detect or classify objects of interest based on their partial views.

In the Local Gradient Smoothing (LGS) [13] method, the authors leverage the observation that within the adversarial patches, the image gradients are usually large due to sharp changes in pixel values within the patches. Gradient smoothing is applied to the image regions exhibiting such a behavior. Their method outperformed other defense methods such as Digital Watermarking, JPEG Compression, Total Variance Minimization, and Feature Squeezing.

The work of [14] observed that near the pixels perturbed by adversarial patches, gradients of classification loss with respect to the input image tend to exhibit large values. This behavior was leveraged along with saliency maps to estimate regions where an adversarial patch may be located. The authors of [14] developed a digital watermarking (DW)-based method to detect such regions with large gradients and mask out these regions from the image. However, in cases of no patch attacks, the saliency map would point towards an object of interest, the processing (and masking) of which may cause reduction in detection performance in clean images. A good defense method is expected not only to reduce the effect of adversarial patches, but also to achieve an accuracy on clean images as close to that achieved without the defense method on clean images.

Some works such as [15] have studied the use of JPEG compression which uses Discrete Cosine Transform (DCT) to eliminate high-frequency components. Although it was shown to defend against the effect of adversarial image perturbations, such methods are not effective against adversarially learnt patches-based attacks. Similarly, works such as [16] which employed Total Variance Minimization, JPEG compression and image quilting, were also found to be ineffective in cases of localized large variations as caused by adversarial patches [13].

The problem with the approaches such as [14] is that they mask out the image regions, causing information loss. If the models are not trained to work with such missing pieces of information, then the clean accuracy also suffers. Contrary to these approaches, we propose a defense method that effectively replaces the suspected attacked regions of an image using its cleaner symmetric half, leveraging the symmetry in vehicles' frontal (or rear) faces. Table 1 provides a summary of the limitations in related works and highlights the contributions of this work.

3 Adversarial Patches-Based Attacks Against VMMR Systems

In this section, we introduce adversarial patches that could be printed and placed on or around vehicles to fool the VMMR surveillance systems. We describe in detail the process of learning these adversarial patches. Unlike the prior works such as [2, 5, 8], our work is the first, to the best of our knowledge, that targets VMMR systems. Unlike stop signs or persons, vehicles not only differ in appearance (multiplicity) but have many similarities as well (inter-class and intra-class ambiguity). In brief, we learn adversarial patches which when placed on or around a vehicle lowers the VMMR classification accuracy.

The adversarial patches-based attacks against VMMR systems could be launched by physically placing the printed patches on the vehicles or by digitally placing the patches on the captured images. The digital placement of patches could be achieved by compromising the surveillance camera network, e.g., through video injection attacks or man-in-the-middle type of attacks. Many works have studied the problem of network intrusions in IoT and camera networks (e.g., [17–29]). The focus of this paper is on defending against attacks launched by placing the adversarial patches on the vehicles regardless of whether it is done physically or digitally. Nonetheless, in training the adversarial patches, the patch transformation and update module (as

Table 1 Summary of related works

Work	Description	Possible overhead	Adversarial patches
Abdel-Hakim [1]	Object detector has to be executed on multiple patches per input image; May require the classifier to be re-trained to classify objects based on partial views.	High	✓
Hayes [14]	Regions with large classification loss gradients are masked out; In absence of adversarial patches, objects of interest may be wrongly masked out, possibly causing loss of information.	Low	✓
Das et al. [15]	Uses JPEG compression to overcome adversarial pixel perturbations; Different regions of an image are compressed using random compression levels at test time; Not effective against adversarial patches-based attacks.	Low - Medium	×
Guo et al. [16]	Before an image is fed to a classifier, image transformations such as bit-depth reduction, JPEG compression, total variance minimization, and image quilting are applied; Ineffective against localized large variations such as those caused by adversarial patches.	Low - Medium	×
Our Proposed Method	Eliminates adversarial patches by using the 'cleaner' symmetric image half; Reduces information loss that occurs if regions are simply masked out; Avoids classifier modification or re-training; Lightweight, adding minimal processing time overhead.	Low	✓

described in Section 3.2) factors into account real-world considerations of physical patch placement and appearance.

In what follows, we describe the method to train and learn such adversarial patches. The overview of the process to learn the adversarial patches is shown in Fig. 2. The main modules are: *Patch Generation and Update*, *Patch Transformation and Placement*, *Model Execution*, *Loss Calculation and Backpropagation*, as described further below. Table 2 describes the main notations used in the paper.

3.1 Patch Generation and Update

The adversarial patch learning process starts with a generated patch that gets updated through the learning process. The initial patch may either be generated with random values or an adversarial patch trained against another object detector (e.g., person detector) could be used. The patch produced by the former approach is termed as a *random initial patch*, whereas the patch used in the latter approach is referred to as a *pre-trained initial patch*. Regardless of the approach to generate the initial patch, the patch undergoes updates through the training process, updating its values based on the back-propagated gradients (that are described in Sect. 3.4).

3.2 Patch Transformation and Placement

The initial and updated patches are transformed and placed on input images in this module. The transformations include scaling, rotation, brightness/contrast adjustments, and noise addition. These transformations have to be such that it is possible to perform gradient computations during backpropagation to update the patch values [2]. The transformed patches are then placed on the input images at specified

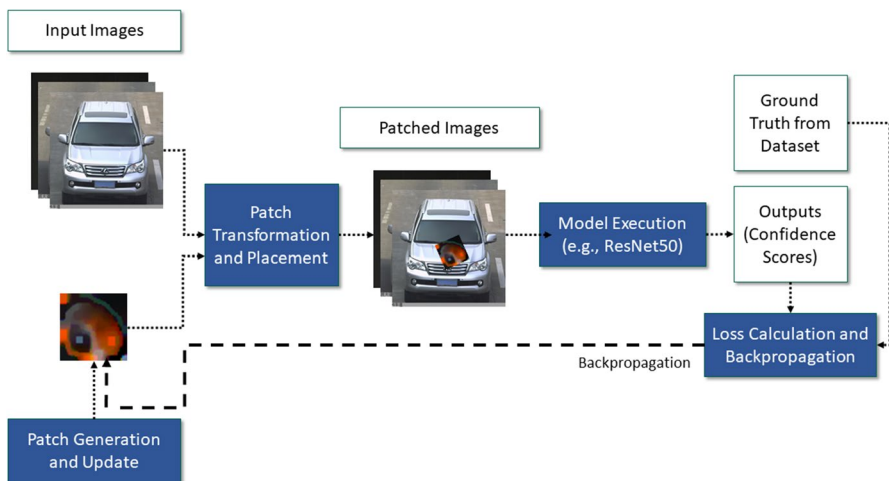


Fig. 2 (Color online) Overview of the adversarial patch learning process targeting a CNN-based VMMR System

Table 2 Table of notations

Symbol	Description
\mathcal{L}_{CC}	The highest confidence score in a list of outputs from the model execution module
\mathcal{L}_{PNP}	Non-printability loss of a learnt patch
\mathcal{L}_{PS}	Patch smoothness loss
\mathcal{L}	The overall loss function (Equation 3)
$I_{i,j}$	Refers to a pixel at the coordinate (i, j) of image I
I_L and I_R	The left and right halves of an image I , respectively
$TV(I)$	The total variation [31], given an image I , as expressed in Equation 4
‘-TL’	Patch placed at the top-left of images in training
‘-RL’	Patch placed at random locations on images in training
‘(TL)’	Patch placed at the top-left of images in testing
‘(RL)’	Patch placed at random locations on images in testing

or random locations in the image. We study and evaluate, in Sect. 6.1, both patch placement approaches (specified or random) and their effectiveness in reducing the VMMR performance.

The patch transformations help in learning adversarial patches that are robust to inevitable factors experienced in real-world applications. An example real-world application is a camera-based VMMR system that captures the images of incoming or exiting vehicles and the adversarial patches may be placed on or around these vehicles.

To elaborate, when a printed adversarial patch is placed on the vehicle, its appearance to the camera may change due to changing conditions such as lighting. Since the vehicles may be at different viewing angles to the camera, the viewing angles of the patches may vary as well. In addition, since vehicles in captured images vary in size, the relative sizes of patches and vehicles vary. Moreover, there may be noise or blur introduced by the camera capturing the input images [2]. Also, an adversary may place the patches at different locations on or around the vehicle. Hence, the patch transformation and placement module incorporates the influences of such factors in the patch training process, thereby achieving robustness to these factors.

3.3 Model Execution

In this module, a deep learning model (CNN) such as ResNet50 [30] that has been trained to achieve high classifications rates for VMMR is used. The patched images from Patch Transformations and Placement module are fed into the Module Execution module (as depicted in Fig. 2). It is worth mentioning that any deep learning model that produces a list of confidence scores could be utilized in place of ResNet50. We choose ResNet50 for VMMR due to its demonstrated success in achieving high accuracy levels for image classification [30].

Each patched image makes a forward pass through the model, producing a list of confidence scores for each of the classes in the considered dataset. The class winning

the highest confidence score that is above a certain acceptable threshold is regarded as the predicted class of the vehicle in the input image. It is worth noting that an attacker doesn't need to know the internal details or architecture of the CNN or other network performing the VMMR. The adversarial patches are trained based on the confidence scores output by the VMMR model.

3.4 Loss Calculation and Backpropagation

The loss calculation module determines the values for the loss functions described below, using the ground truth data of input images and the outputs from the model execution module. In this work, we adopt the following loss functions, inspired by the approach of [2]: (i) Class Confidence Loss (\mathcal{L}_{CC}), (ii) Non-Printability Loss (\mathcal{L}_{PNP}), and (iii) Patch Smoothness Loss (\mathcal{L}_{PS}). The three loss functions are described below.

- \mathcal{L}_{CC} : The highest confidence score in a list of outputs from the model execution module. The goal is to minimize the confidence score corresponding to the ground truth class to achieve the objective of the adversarial patch, i.e., to miss-classify the given vehicle's image as any other vehicle class.
- \mathcal{L}_{PNP} : The non-printability loss measures how far the adversarial patch's pixel color values are from the given set of commonly printable colors [8]. It is formulated as:

$$\mathcal{L}_{PNP} = \sum_{p_{i,j} \in P} \min_{c \in C} |p_{i,j} - c| \quad (1)$$

Here, c refers to a color from the set of commonly printable colors given by C whereas $p_{i,j}$ refers to a pixel at (i, j) -th coordinate in the adversarial patch P .

- \mathcal{L}_{PS} : The patch smoothness loss measures the total variation [31] of the patch. Lower the total variation, higher would be the patch smoothness and vice-versa. The goal in this work is to learn adversarial patches that are smoother, following the approach of [2, 8], so that the patched images resemble closely the natural images in terms of smoothly gradually changing colors. Additionally, as noted in [8], cameras may not be able to properly capture high variations in a patch's adjacent pixels due to sampling noise. Hence, patches with low smoothness loss, i.e. high smoothness, are more effective for real-world applications. The \mathcal{L}_{PS} is formulated as:

$$\mathcal{L}_{PS} = \sum_{p_{i,j} \in P} (p_{i,j} - p_{i+1,j})^2 + (p_{i,j} - p_{i,j+1})^2 \quad (2)$$

The overall loss function is a weighted aggregate of the above three loss functions, as given below, where the values of α , β and γ are chosen through empirical evaluations:

$$\mathcal{L} = \alpha \mathcal{L}_{CC} + \beta \mathcal{L}_{PNP} + \gamma \mathcal{L}_{PS} \quad (3)$$

The loss function is minimized through backpropagation and Adam optimizer [32]. In doing so, only the patch values are updated while the weights of the deep learning model (e.g., ResNet50) are frozen.

4 Proposed Defense Method: Symmetric Image-Half Flip and Replace (SIHFR)

In this section, we introduce the proposed *Symmetric Image-Half Flip and Replace (SIHFR)* method designed and developed to defend against adversarial patches-based attacks on VMMR surveillance systems. The proposed method is based on certain observations about the nature of images captured by surveillance cameras in VMMR systems. Specifically, in this work we consider VMMR systems that capture images of incoming or outgoing vehicles such that the captured images have one vehicle each. We assume that the images which are going to be fed to VMMR algorithms have the vehicles at their centers, i.e., the vehicles are centrally aligned with respect to the input images. This is a reasonable assumption since such images are commonly found in VMMR deployment scenarios such as checkpoints, entrances or exits of secured areas [6, 33]. In fact, the dataset used in this work was collected using a real-world surveillance camera capturing incoming vehicles, and contains images of the aforementioned nature. In the current work, we focus on frontal view images only, however our method is applicable to rear view images as well.

The SIHFR method has been designed to meet the following requirements. First, it shall not require re-training or fine-tuning of the CNN-based VMMR models. This is essential because re-training and fine-tuning of CNN-based VMMR models is an expensive process in terms of computational and time costs. Second, it shall eliminate or reduce the effect of adversarial patches on the VMMR scores. Third, it shall be lightweight and shall not add significant overheads in terms of computational or time resources, both in training and execution phases.

The working principle of the proposed SIHFR method is based on the following observations. The vehicle images captured in the aforementioned conditions are found to be symmetric around a vertical central axis, hereby referred to as the central axis, even under slight variations in viewpoints of the incoming vehicles, as illustrated in Fig. 3. An adversary could place the adversarial patch anywhere on the vehicle such that the patch appears on the left or right of the central axis of symmetry. It may also be possible that the patch is placed along the central axis such that it partly appears on both sides of the axis.

Since the left and right image halves (split by the central axis) are symmetric to each other, they exhibit similar values of Total Variation (TV) [31], under normal conditions, i.e., in absence of any adversarial patches. Given an image (or image part) I , the value of total variation, $TV(I)$, is calculated as per Equation 4, where $I_{i,j}$ refers to a pixel at the coordinate (i, j) of I . When an adversarial patch appears on either image half, the TV value of that half will be different from that of the other image half. We utilize this difference in TV to identify the image half that is 'clean' and the one that is potentially 'modified' with the adversarial patch. The clean half is selected, horizontally flipped and then copied onto the other half. In this manner, the method gets rid of the adversarial patch which was placed on the 'modified' image half. Figure 3 presents a flowchart of the proposed method with a sample clean image (with no adversarial patch) and an attacked image (with an adversarial patch).

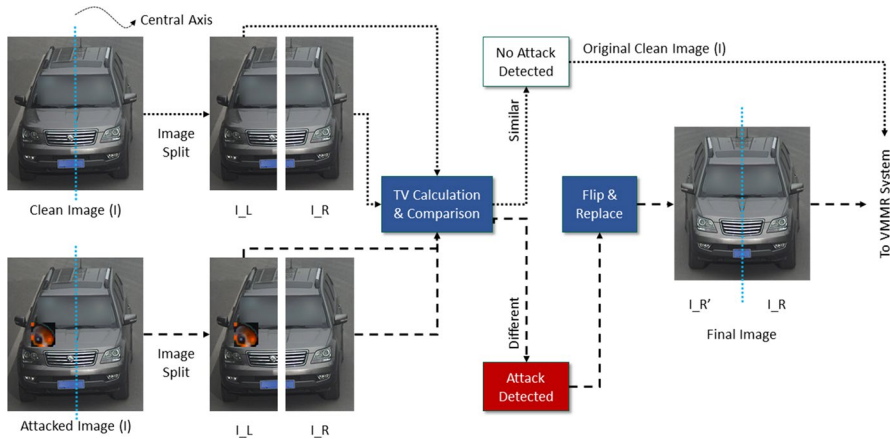


Fig. 3 (Color online) A flowchart of the proposed SIHFR method to defend against adversarial patches-based attacks on VMMR systems. In the final image, $I_{R'}$ refers to the horizontally flipped I_R which was selected to replace I_L because $TV(I_L) > TV(I_R)$

The ‘TV Calculation and Comparison’ step (Fig. 3) calculates and compares the TV values of the left and right image halves I_L and I_R , respectively. If the TV values are similar, no adversarial patches are detected and so the original input image is passed on to the VMMR system. On the other hand, if the TV values are different, an adversarial patch attack is detected. In the ‘Flip and Replace’ step, the image half with a lower TV value is selected as the clean half which is flipped to replace the attacked half. This is based on the observation, using training images, that the image half with an adversarial patch has a higher TV value, despite the fact that the adversarial patch training takes into account the goal of maximizing patch smoothness (as described in Sect. 3.4) to reduce the total variation.

$$TV(I) = \sum_{I_{ij} \in I} (I_{ij} - I_{i+1,j})^2 + (I_{ij} - I_{i,j+1})^2 \tag{4}$$

5 Experimental Setup

In this section, we present the experimental setup used in training and testing evaluations of adversarial patches targeted against VMMR systems and of the proposed SIHFR defense method. Beginning with a description of the real-world dataset used in this work and describing the performance metrics that shall be used in the evaluations, we then present the training process of three groups of adversarial patches developed and evaluated in this work.

5.1 Dataset

In order to evaluate the effectiveness of adversarial patches against VMMR and of the SIHFR defense method, we choose the CompCars Surveillance Dataset [34]. Though there are other publicly available datasets for VMMR [6], we found [34]'s dataset to be most representative of the real world scenarios this work aims to target [6]. Its composition is such that it allows one to train a VMMR model that deals with the challenges of multiplicity and inter- or intra-class ambiguities. The adversarial patches developed in this work are targeted against such a robust VMMR system.

The images in CompCars Surveillance Dataset were collected by on-road surveillance cameras that capture the oncoming vehicles' frontal views. The images were taken under varying lighting conditions. The dataset has 281 classes with a total of 31, 149 images for training and 13, 334 images for testing. It also contains make-model classes that have different versions and appearances over different years.

In real-world applications, a VMMR system could encounter different types of vehicles. As such, the selected dataset comprises of the following different vehicle types: sedan, hatchback, fastback, SUV, MPV, minibus, estate, crossover, convertible, hardtop convertible, sports, and pickup [34].

5.2 Performance Metrics

The performance of a VMMR system, in terms of how good or bad it is in classifying different make-model classes, is assessed based on the following metrics: precision, recall, and F1 scores, averaged across all classes in the dataset. The effectiveness of an adversarial patch in fooling the VMMR system could then be assessed by the amount of reduction it causes in these VMMR scores. The higher the reduction, more effective is the adversarial patch. Amongst these scores, reduction in recall scores is the most important for a non-targeted attack [2].

Below, we provide classwise definitions for these metrics, where TP_i , FP_i and FN_i refer to the True Positives, False Positives and False Negatives with respect to a class- i , respectively. The classwise metrics are averaged to obtain average precision, recall and F1 scores that represent the overall VMMR performance.

- Attack Detection Rate: the ratio of correctly detected adversarial patches-based attacks to the total number of attacked test images.
- Precision: For a class i , its Precision score P_i is:

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (5)$$

- Recall: For a class i , its Recall score R_i is:

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad (6)$$

- F1-Score: a harmonic average of Precision and Recall scores

$$F1\text{-Score}_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \quad (7)$$

Then, a reduction in these metrics (especially the Recall metric) caused by placement of the proposed adversarial patches on the images is measured in terms of a percentage decrease that reflects the attack's success rate. The success of a defense method on the other hand is assessed based on the attack detection rate and on the improvements in scores of the attacked VMMR system.

5.3 Training of Adversarial Patches

The adversarial patches are learnt following the procedure described in Sect. 3. In this work, we study three groups of adversarial patches which differ in the number of epochs they're trained for, or in the initial patch configuration, or in the weights used in the loss function of Equation 3.

5.3.1 Group 1

In the first group, we train three patches, setting the weights $\alpha = \beta = \gamma = 1$ for the loss function of Equation 3 and starting from a blank gray initial patch. The three patches P1, P2 and P3 differ in the number of epochs they're trained for:

- P1: trained for a lower number of epochs (45) and smaller batch size (25).
- P2: trained for a higher number of epochs (200) and a larger batch size (100).
- P3: trained for a higher number of epochs (300) and a large batch size (100).

The patches P2 and P3 are trained under two patch placement settings: (i) *TL*: the placement of patches is at a fixed location on input images, i.e., the top-left, and (ii) *RL*: the patches are placed at random locations on the input images. In testing, we evaluate placement of patches at fixed (top-left) and random locations. Correspondingly, the suffixes '-TL' and '-RL' to the patch names shall refer to the placement setting in training while the suffixes '(TL)' and '(RL)' refer to the placement settings used in testing.

The patch P1, two versions of P2, and two versions of P3 are shown in Fig. 4. In this work, we choose a patch size of 50×50 (before transformations) and input images are resized to 224×224 . In Fig. 5, we show the curves for \mathcal{L}_{CC} , \mathcal{L}_{PNP} , and \mathcal{L}_{PS} obtained during training of P2-RL and P3-RL. While P2-RL was extracted from epoch 200, P3-RL was extracted from epoch 300. The values of all three loss functions are higher at epoch 200 (P2-RL) than at epoch 300 (P3-RL), hinting that P3-RL could be more effective than P2-RL in reducing the VMMR scores.

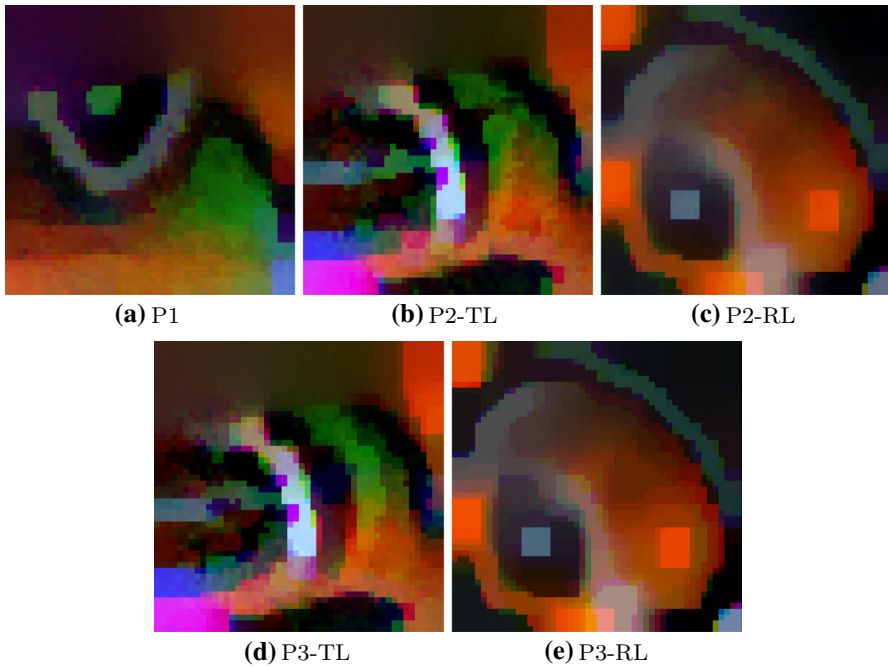


Fig. 4 (Color online) First group of adversarial patches (P1, P2, P3) developed in this work to fool a ResNet50-based VMMR system

5.3.2 Group 2

The second group of patches we developed are trained with assigning a higher weight to the class confidence loss L_{CC} in Equation 3 by setting $\alpha = 10$. A higher weight for L_{CC} would force the learning process to give more importance to reducing class confidence scores of the target VMMR model than the other components

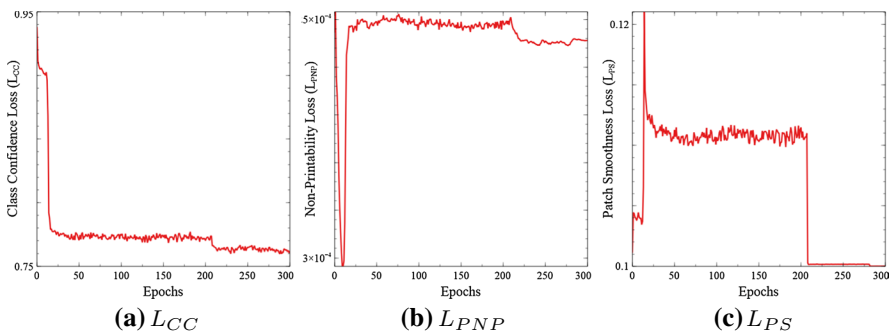


Fig. 5 Training of Adversarial Patches P2-RL and P3-RL: Showing the L_{CC} , L_{PNP} and L_{PS} over 300 epochs. P2-RL was extracted from epoch 200 whereas P3-RL was extracted from epoch 300

of the loss function. We train three patches in this group, with batch sizes of 256 per epoch and patches placed at random locations in training (RL patch placement setting):

- *P4A*: trained for 300 epochs
- *P4B*: trained for 400 epochs
- *P4C*: trained for 500 epochs.

Figure 6 shows the three patches in this group. As one could observe on comparing them with the first group of patches, the P4 patches are less smoother than P1, P2, P3. This is due to the fact that the learning process for P4 (A,B,C) gives lesser importance to the patch smoothness loss L_{PS} . Moreover, there is hardly any visual difference between P4B and P4C, though there are slight changes in the numerical RGB values, hinting that the success of attacks using P4B and P4C may be similar.

Looking at the training curves of P4s (Fig. 7), one could observe that values of three loss terms (L_{CC} , L_{PNP} and L_{PS}) do not decrease much beyond epoch 400 which explains the close similarity of P4B and P4C besides hinting that training for more epochs may not result in any drastically better adversarial patches.

5.3.3 Group 3

In the third group, adversarial patches trained to fool other object detectors are used as the initial patches in our training process. This can effectively be seen as fine-tuning of adversarial patches (that were pre-trained to attack another application) in order to learn adversarial patches to fool VMMR. In this work, we utilize the patches learnt in [2] as initial patches in learning patches P5A, P5B, and P5C to fool VMMR.

The work of [2] had trained three kinds of adversarial patches, differing in the goal of the optimization process: (a) a patch that is trained to minimize the confidence score of class ‘person’ as well as objectness score of the targeted model, (b)

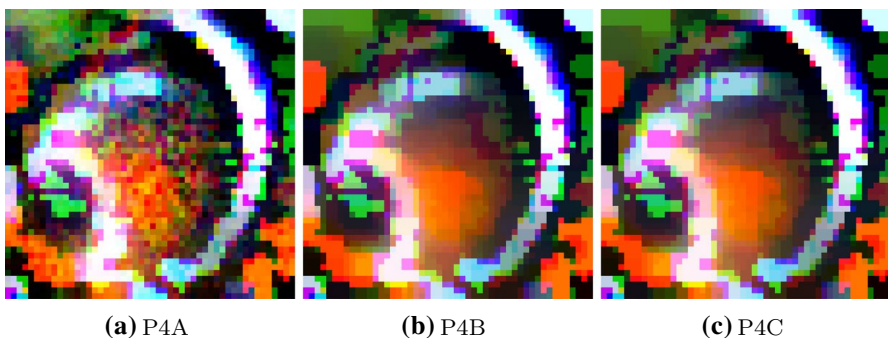


Fig. 6 (Color online) Second group of adversarial patches (P4 Series) developed in this work to fool a ResNet50-based VMMR system

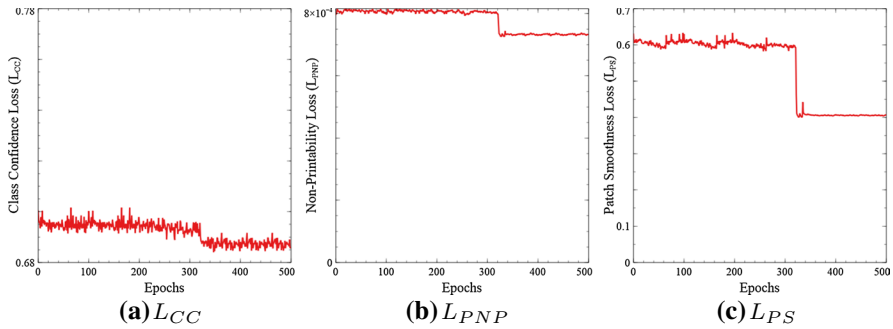


Fig. 7 Training of Adversarial Patches P4A/B/C: Showing the L_{CC} , L_{PNP} and L_{PS} over 500 epochs. P4A, P4B and P4C were extracted from epochs 300, 400, and 500, respectively

a patch that is trained to minimize only the objectness score and (c) a patch that is trained to minimize only the confidence score of the ‘person’ class. The objectness score is more generic than class confidence score and refers to the probability that there exists an object of interest (regardless of which class it belongs to) in the image at some location. Correspondingly, the three patches we train in Group 3, from these initial patches, are referred to as P5A, P5B, and P5C. Figure 8 shows the training curves indicating the loss values obtained during training these patches for 100 epochs. The three resulting patches are shown in Fig. 9.

6 Results & Discussions

In this section, we conduct experiments to investigate the following: (a) performance evaluation of the developed adversarial patches in reducing VMMR scores and performance comparison of these patches, (b) impact of patch placement location on attack effectiveness, and (c) evaluating the effectiveness and robustness of the proposed SIHFR defense method and comparing it against a state-of-the-art defense method.

6.1 Evaluation of Developed Adversarial Patches

The following subsections present the evaluations of the three groups of adversarial patches developed in this work.

6.1.1 Evaluating Group 1 Patches (P1, P2, P3)

We evaluate P1 under two settings: (a) TL: P1 placed at a fixed location (top-left) on the test images, and (b) RL: P1 placed at random locations on the test images. Table 3 summarizes the results of evaluating P1 against a ResNet50-based VMMR. The suffix ‘-TL’ to the patch name refers to the top-left placement setting in training

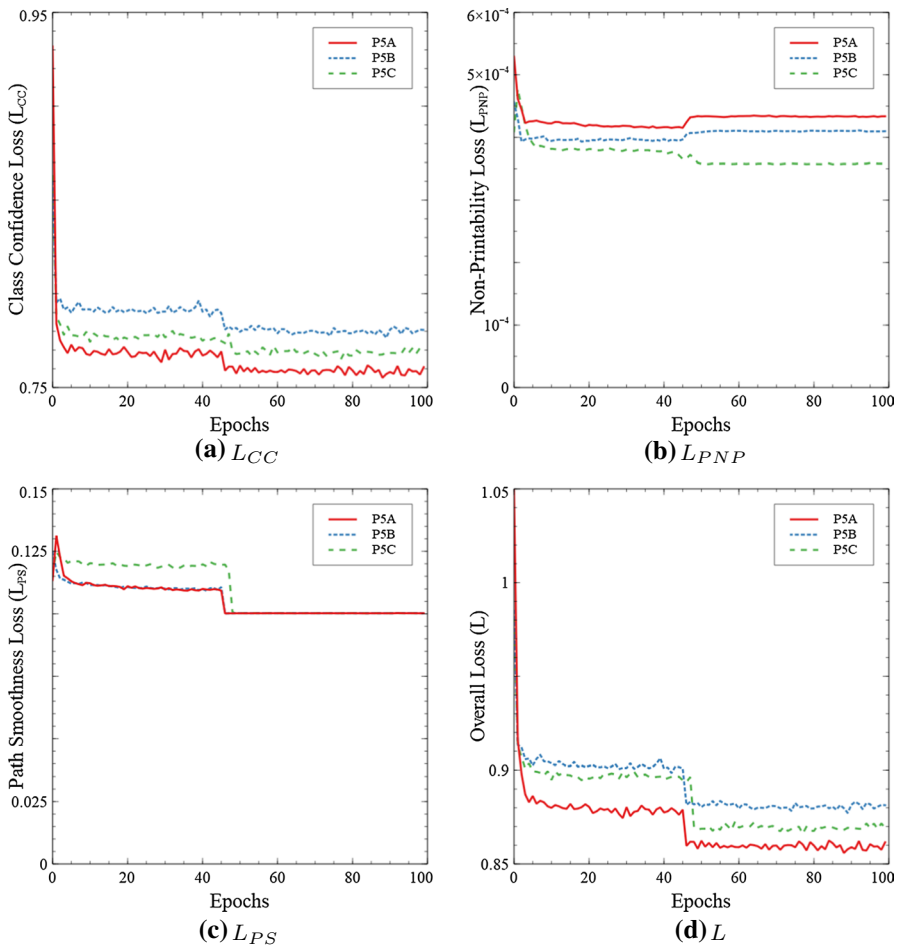


Fig. 8 Training of Adversarial Patch P5A/B/C: Showing the L_{CC} , L_{PNP} , L_{PS} and the overall loss L over 100 epochs

while the suffixes ‘(TL)’ and ‘(RL)’ refer to the top-left and random placement settings, respectively, used in testing.

Analyzing the results of Table 3, we find that, P1, learnt by fixed placement (at top-left) of training images, achieved the best reduction (though very small) in VMMR precision, recall and F1 scores when it was placed at top-left in test images as well (see P1-TL (TL) in Table 3). The learnt P1 when placed at random locations on test images achieved a very slight reduction in VMMR performance metrics. Moreover, we observe that learnt patches are more effective than random patches in reducing the VMMR system’s classification performance.

To evaluate P2, we train two versions of P2, namely: P2-TL and P2-RL. While P2-TL is trained through placement at a fixed location (top-left) on training images, P2-RL is trained through placement at random locations on training

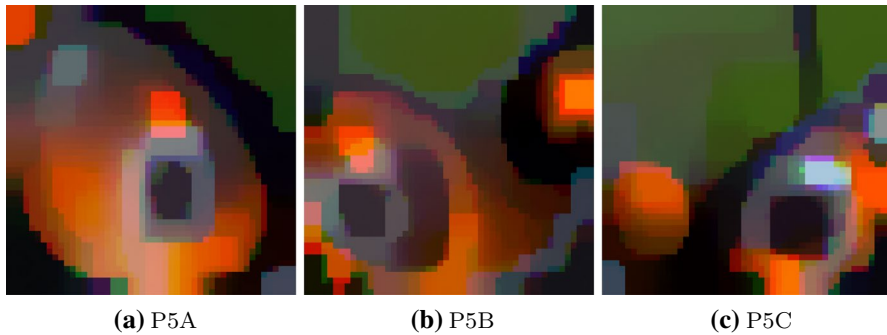


Fig. 9 (Color online) Third group of adversarial patches (P5 series) developed in this work to fool a ResNet50-based VMMR system, using the patches learnt by [2] as starting points followed by fine-tuning on the VMMR dataset

Table 3 Evaluation results of P1

Patch	Avg. precision	Avg. recall	Avg. F1 Score
None	0.9875	0.9793	0.9821
P1-TL (TL)	0.9780	0.9639	0.9682
P1-TL (RL)	0.9841	0.9762	0.9790
Random-TL (TL)	0.9872	0.9797	0.9827
Random-TL (RL)	0.9865	0.9780	0.9811

Table 4 Evaluation Results of P2-TL

Patch	Avg. precision	Avg. recall	Avg. F1 Score
None	0.9875	0.9793	0.9821
P2-TL (TL)	0.9772	0.9699	0.9717
P2-TL (RL)	0.9733	0.9662	0.9679
Random-TL (TL)	0.9854	0.9776	0.9805
Random-TL (RL)	0.9861	0.9787	0.9815

images. P2-TL and P2-RL are evaluated under both placement settings TL and RL on test images.

Table 4 presents the results of P2-TL-based attacks against the target VMMR system. We note that P2-TL achieved a better reduction in VMMR precision, recall and F1 scores than the random noise patches. When it was placed at random locations (RL placement setting) in test images, P2-TL achieved slightly better reduction than when placed at the top-left location on test images (TL placement setting).

Evaluating P2-RL, we observe the following from Table 5. The highest decrease in VMMR performance (precision by 9.72%, recall by 36.15% and F1 score by 29.73%) was achieved by placing P2 at random locations both during

training and testing, i.e., by P2-RL (RL). When placed at top-left of test images, P2-RL was able to cause only a slight decrease in the classification scores.

To evaluate P3, as with P2, we train two versions of P3, namely: P3-TL and P3-RL, each evaluated under both TL and RL settings described previously. As we observe the results in Table 6, we find that P3-TL causes very slight reduction in precision, recall and F1 scores of VMMR. When placed at random locations on input test images (RL setting), P3-TL (RL) achieved slightly better reduction in these metrics as compared to that by P3-TL (TL).

Next, we evaluated P3-RL which was trained through placements at random locations on input images. With P3-RL, we note that the best results were obtained under the RL placement settings during testing (see Table 7). Most notably, the average recall rate was brought down to 0.6170 from 0.9793, a decrease of around 37.0%.

6.1.2 Evaluating Group 2 Patches (P4A/B/C)

We evaluate P4A, the patch trained for 300 epochs with $\alpha = 10$ as weight to the L_{CC} component of the loss function of Equation 3. Table 8 summarizes the results. The most effective attack with P4A was under the RL patch placement setting during both training and testing, bringing down the average precision, recall and f1-scores

Table 5 Evaluation results of P2-RL

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P2-RL (TL)	0.9665	0.9595	0.9595
P2-RL (RL)	0.8915	0.6253	0.6901
Random-RL (TL)	0.9859	0.9780	0.9810
Random-RL (RL)	0.9834	0.9757	0.9786

Table 6 Evaluation results of P3-TL

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P3-TL (TL)	0.9771	0.9706	0.9722
P3-TL (RL)	0.9726	0.9652	0.9668
Random-TL (TL)	0.9849	0.9783	0.9808
Random-TL (RL)	0.9832	0.9750	0.9781

Table 7 Evaluation Results of P3-RL

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P3-RL (TL)	0.9641	0.9566	0.9563
P3-RL (RL)	0.8974	0.6170	0.6852
Random (TL)	0.9855	0.9779	0.9807
Random (RL)	0.9818	0.9738	0.9769

to 0.8836, 0.5242, 0.5978, respectively. The P4A-RL (RL) achieved 0.4063 points lower average recall than by the random noise patches under the same patch placement settings (RL) and 0.4551 points lower average recall than the case with no adversarial patches-based attacks.

Next, we evaluate if training P4A for more number of epochs, yielding P4B (at 400 epochs) and P4C (at 500 epochs), could achieve better success in reducing the VMMR scores. Tables 9,10 summarize the results. The patch P4B, under RL patch placement setting in training and testing, reduced the VMMR average recall score to 0.5064, a 0.4729 points reduction from the no attacks case. The average precision and f1 scores were down by 0.1032 and 0.3990 points, respectively.

The patch P4C was also most effective under the RL patch placement setting in training and testing phases. The average precision, recall and f1 scores were reduced by 0.1049, 0.4740 and 0.4044 points respectively, in comparison to the VMMR scores in absence of any adversarial patches.

When we compare the performance of P4A, P4B, P4C, we find that P4C achieved 0.0011 and 0.0189 points lower average recall score than P4B and P4A, respectively. The gains of training P4C for 100 more epochs than P4B were minimal.

With all three versions of P4, we find that the patch placement at top-left of the input images during testing achieved minimal success in lowering the VMMR

Table 8 Evaluation results of P4A

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P4A-RL (TL)	0.9498	0.9349	0.9331
P4A-RL (RL)	0.8836	0.5242	0.5978
Random (TL)	0.9860	0.9782	0.9811
Random (RL)	0.9504	0.9305	0.9362

Table 9 Evaluation results of P4B

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P4B-RL (TL)	0.9469	0.9310	0.9282
P4B-RL (RL)	0.8843	0.5064	0.5831
Random (TL)	0.9853	0.9776	0.9805
Random (RL)	0.9514	0.9335	0.9388

Table 10 Evaluation results of P4C

Patch	Avg. precision	Avg. recall	Avg, F1 Score
None	0.9875	0.9793	0.9821
P4C-RL (TL)	0.9472	0.9318	0.9289
P4C-RL (RL)	0.8826	0.5053	0.5777
Random (TL)	0.9859	0.9778	0.9809
Random (RL)	0.9523	0.9332	0.9383

Table 11 Evaluation results of P5A

Patch	Avg. precision	Avg. recall	Avg. F1 Score
None	0.9875	0.9793	0.9821
P5A-RL (TL)	0.9655	0.9564	0.9571
P5A-RL (RL)	0.9057	0.6007	0.6770
Random (TL)	0.9863	0.9784	0.9814
Random (RL)	0.9521	0.9324	0.9378

Table 12 Evaluation results of P5B

Patch	Avg. precision	Avg. recall	Avg. F1 Score
None	0.9875	0.9793	0.9821
P5B-RL (TL)	0.9728	0.9647	0.9662
P5B-RL (RL)	0.9124	0.6762	0.7433
Random (TL)	0.9855	0.9779	0.9807
Random (RL)	0.9537	0.9330	0.9389

scores in comparison to the success achieved by the patches under random location (RL) patch placement strategy. While P4A-RL (TL) lowered the average recall by 0.0444 points, P4B-RL (TL) and P4C-RL (TL) lowered it by 0.0483 and 0.0475 points respectively, when compared to the scores obtained in absence of any adversarial patches. In contrast, these patches achieved around 10× more reduction under the random location patch placement strategy in testing (denoted by the patch name and suffix ‘(RL)’ in the respective Tables).

6.1.3 Evaluating Group 3 Patches (P5A/B/C)

There are three versions of P5, as described in 5.3.3: P5A, P5B, and P5C. Tables 11, 12, and 13 summarize the evaluations results respectively. The three patches were fine-tuned (starting from the three patches of [2]) and trained by randomly placing the patches on training images (i.e., under the RL patch placement setting).

On the testing dataset, the patch P5A achieved a reduction in average recall scores by 0.0229 points under the TL patch placement strategy and by 0.3786 points under the RL patch placement strategy, the latter achieving 37.2% more success than the former in reducing the average recall score.

Looking at the performance of P5B, we find that it reduced the average precision, recall and f1 scores of VMMR by 0.0751, 0.3031, and 0.2388 points respectively, following the RL patch placement strategy. The reduction in average recall achieved by P5B-RL (TL), i.e. with TL patch placement strategy on testing images, was only by 0.0146 points.

The patch P5C achieved best results when placed at random locations on test images, i.e., following the RL patch placements strategy in testing. It achieved a reduction in average precision, recall and f1 scores by 0.0796, 0.2660 and 0.2135

Table 13 Evaluation results of P5C

Patch	Avg. precision	Avg. recall	Avg. F1 Score
None	0.9875	0.9793	0.9821
P5C-RL (TL)	0.9708	0.9613	0.9627
P5C-RL (RL)	0.9079	0.7133	0.7686
Random (TL)	0.9862	0.9780	0.9811
Random (RL)	0.9440	0.9301	0.9334

points respectively when compared against the case of no adversarial patches-based attacks. With P5C as well, following the TL patch placement strategy on testing images yielded minimal success in reducing the VMMR scores. The average recall score reduction achieved by P5C-RL (TL) was only by 0.0180 points.

Upon comparing P5A, P5B and P5C based on their best results in Tables 11, 12, and 13, we make the following observations. The patch P5A was most successful in reducing the VMMR scores, followed by P5B and P5C. P5A yielded 38.7% reduction in average recall scores, whereas P5B and P5C achieved 31% and 27.2% reduction in average recall, compared to that obtained under no adversarial patches-based attacks. This indicates that the patch of [2] that was trained to minimize class confidence score as well as objectness score serves as a better initial patch for fine-tuning than their other two patches which minimized either the class confidence score or the objectness score alone. The three patches of [2] were used in the fine-tuning process to develop patches P5A, P5B and P5C respectively, as described in Sect. 5.3.3.

From the above evaluations, we observe the following. First, trained adversarial patches are more effective than random noise patches in reducing the classification performance of VMMR. Second, assigning a higher weight to the L_{CC} component of the loss function of Equation 3 results in adversarial patches that are not as smooth as patches learnt with equally weighted components of the loss function. For example, the patches P4A,B,C are less smoother than patches P1,P2, or P3 (compare Figs. 4 and 6). Third, the trained adversarial patches are more effective in fooling the VMMR system when placed at random locations on the test images, i.e., under the RL patch placement strategy, than when placed at the top-left location on test images.

6.2 Comparison of Developed Adversarial Patches

To compare the effectiveness of the developed patches we choose to look at the reduction in average recall score of the targeted VMMR system as it most representatively reflects the success of an adversarial attack [2]. In Table 14, we present the reduction in recall scores achieved by the best settings for each patch, compared against the baseline average recall of VMMR when no adversarial patches are present.

The higher reduction in recall caused by P3-TL (RL) vs. P2-TL (RL) and P3-RL (RL) vs. P2-RL (RL) indicates, as expected, that an adversarial patch trained for more number of epochs is more effective, given the same strategy of

placing the patches. Amongst the patches P1, P2 and P3, the best reduction in recall, of almost 37.0%, is achieved by P3-RL (RL), indicating that placing the adversarial patches at random locations on images during training and execution is most effective in fooling the ResNet50-based VMMR system.

Amongst the patches of all three groups, we find that the best results were achieved by P4C-RL (RL) which reduced the average recall score (by 48.40%). On the other hand, amongst the patches tested with the TL patch placement strategy, P4B-RL (TL) achieved the best reduction in recall score (by 4.93%).

To answer the question “Whether adversarial patches learnt from fine-tuning of patches pre-trained for another application are more effective than those learnt from scratch targeting the VMMR application?”, we compare the results of the patches P4 and P5. The best reduction in VMMR scores achieved by P5 was with P5B-RL (RL), bringing down the average precision, recall and F1 scores by 7.61%, 30.95% and 24.32% respectively. In contrast, the best reduction in scores achieved by P4 was by 10.62%, 48.40%, and 41.18% respectively, with P4C-RL (RL). So, P4 achieved 3.01, 17.45, 16.86 percentage points higher reduction in the average VMMR precision, recall and f1 scores respectively, in comparison to that by P5. This indicates that training adversarial patches from scratch could be more effective against ResNet50-based VMMR models than fine-tuning patches pre-trained to attack models trained for other applications such as person detection. In addition, it indicates that a higher weight to the \mathcal{L}_{CC} during training (e.g., of P4) yields more effective adversarial patches.

Table 14 Comparing effectiveness of the developed adversarial patches

Patch/setting	Avg. recall	Reduction (%)
No Patch	0.9793	–
P1-TL (TL)	0.9639	1.57
P2-TL (RL)	0.9662	1.34
P2-RL (RL)	0.6253	36.15
P3-TL (RL)	0.9652	1.44
P3-RL (RL)	0.6170	37.0
P4A-RL (TL)	0.9349	4.53
P4A-RL (RL)	0.5242	46.47
P4B-RL (TL)	0.9310	4.93
P4B-RL (RL)	0.5064	48.29
P4C-RL (TL)	0.9318	4.85
P4C-RL (RL)	0.5053	48.40
P5A-RL (TL)	0.9564	2.34
P5A-RL (RL)	0.6007	38.66
P5B-RL (TL)	0.9647	1.49
P5B-RL (RL)	0.6762	30.95
P5C-RL (TL)	0.9613	1.84
P5C-RL (RL)	0.7133	27.16

6.3 Impact of Patch Placement Location on Attack Effectiveness

Based on the results from the previous experiments, we learnt that placing the patches at random locations on the input images, both during training and testing, yields the most effective attacks. Next, we investigate which regions of the input image are more effective than others for the random patch placement.

We study five regions in these experiments, as depicted in Fig. 10: (i) R1, the top-right quadrant, (ii) R2, the top-left quadrant of the input image, (iii) R3, the bottom-left quadrant, (iv) R4, the bottom-right quadrant, and (v) R5, a central region on the input image focusing on the vehicle's face. In each region, the patches are randomly placed on the input images, constrained by the region boundaries. For example, patches to be placed in R1 cannot be placed on any location in the top-left quadrant (R2). In these experiments, we utilize the P4C-RL patch which achieved the best results in the previous experiments.

Given a patch size of 50×50 and input image size of 224×224 , the patch width (height) is around 22% of the image width (height). So, the patches in each of the five regions have their center coordinates (x, y) restricted by certain factors of the input image width and height, as presented in Table 15, to ensure that patches are contained within their intended quadrants. For example, a patch in R1 could have a minimum (x, y) coordinate of $(0.5 + W_p/W_I, 0.0 + W_p/W_I)$ and a maximum $(x,$

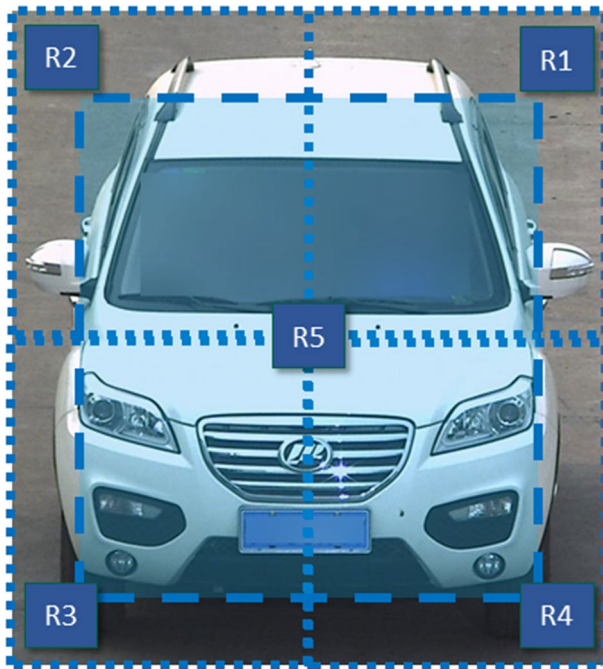


Fig. 10 Demarcating the five regions used in Sect. 6.3 to investigate the impact of patch placement location on effectiveness of adversarial attacks against the target VMMR system, using the CompCars dataset [34]

y) coordinate of $(1.0 - W_p/W_I, 0.5 - W_p/W_I)$, where W and H represent width and height, while subscripts I and P refer to input images and patches respectively. In Fig. 11, we show some samples of the vehicle images attacked with adversarial patches placed randomly within each of the regions R1-R5 respectively.

In Table 16, we present the evaluation results of placing the adversarial patch in each of the five regions. The lowest reduction in average recall score for VMMR was obtained when the patch was placed in R4 while placement of the adversarial patch in R1 yielded the second lowest reduction in average recall score. This indicates that placement of adversarial patches in R1 and R4 is not the most effective approach of an attack. A reason for this could be that these regions of a vehicle's image are not the most informative or important ones in a CNN trained for VMMR, possibly due to limited discriminatory features occurring in R1.

From Table 16, we find that the most effective region to place the adversarial patch P4C was R5, achieving a reduction of 10.67%, 50.26% and 43.78% in the average precision, recall and f1 scores of the VMMR system. Similarly, the adversarial patch P5A also performed most effective in region R5, achieving reductions of 8.02%, 33.87% and 27.38% in the average scores respectively. Although R5 overlaps



Fig. 11 (Color online) Sample vehicle images attacked with adversarial patches placed randomly within regions R1-R5 respectively (images from CompCars dataset [34])

Table 15 Restrictions on Adversarial Patch Coordinates in the Five Regions

Region	Min. x	Min. y	Max. x	Max. y
R1	$0.5 + W_p/W_l$	$0.0 + W_p/W_l$	$1.0 - W_p/W_l$	$0.5 - W_p/W_l$
R2	$0.0 + W_p/W_l$	$0.0 + W_p/W_l$	$0.5 - W_p/W_l$	$0.5 - W_p/W_l$
R3	$0.0 + W_p/W_l$	$0.5 + W_p/W_l$	$0.5 - W_p/W_l$	$1 - W_p/W_l$
R4	$0.5 + W_p/W_l$	$0.5 + W_p/W_l$	$1.0 - W_p/W_l$	$1.0 - W_p/W_l$
R5	$0.2 + W_p/W_l$	$0.2 + W_p/W_l$	$0.8 - W_p/W_l$	$0.8 - W_p/W_l$

Table 16 Evaluating the impact of patch placement location on P4C's attack effectiveness

Region	Avg. precision	Avg. recall	Avg. F1 Score
No Patch	0.9875	0.9793	0.9821
R1	0.9343	0.8567	0.8732
R2	0.9228	0.8331	0.8485
R3	0.9058	0.7971	0.8181
R4	0.9293	0.8887	0.8939
R5	0.8821	0.4871	0.5521

with R1, R2, R3 and R4, it tightly encloses a vehicle's face unlike the other quadrants that include regions away from the vehicle's face as well. This could be the reason why patches placed randomly within R5 had the most impact in reducing VMMR scores.

6.4 Evaluating the Proposed SIHFR Defense Method

We evaluate the effectiveness of the proposed SIHFR defense method against adversarial patches trained to fool CNN-based VMMR systems. Moreover, we analyse the overhead incurred by the addition of SIHFR module to the VMMR module. We also compare the performance of SIHFR with that of a related state-of-the-art defense method in defending the VMMR system from adversarial patches. In addition, we study the effect of patch size on the robustness of the proposed SIHFR defense method.

6.4.1 Defending Against the Adversarial Patches

In this section, we study robustness of the proposed defense method against the developed adversarial patches. Based on previous experiments (without any defense method in place), the two best adversarial patches were P4C and P5A. The most effective patch placement region was found to be R5. We test the proposed defense method's success in eliminating or reducing the impact of adversarial patches P4C and P5A on the target VMMR system. The patches are placed randomly within the R5 region in separate experiments. Table 17 shows the average recall scores (at 95% confidence intervals) of the target VMMR system with and without SIHFR in the presence and absence of adversarial patches. The target VMMR system with SIHFR

activated experienced a slight decrease in average recall scores when there were no adversarial patches. This is because the target VMMR system which is based on ResNet50 model was not trained using benign images constructed using the symmetric image halves as in SIHFR, but was trained on un-modified images.

On the other hand, the average recall scores of the target VMMR system improved with our proposed SIHFR defense method by 69.28% and 20.25% on average, in the case of P4C- and P5A-based attacks respectively. These results indicate the effectiveness of the proposed SIHFR method in eliminating the influence of adversarial patches on the VMMR system to a significant extent, though there still remains room for further improvement.

6.4.2 Evaluating SIHFR's Overhead on the target VMMR System

In order to evaluate the overhead added to a CNN-based VMMR system by activating the proposed SIHFR defense method, we examine the processing time consumed by the SIHFR module in comparison to the processing time of the CNN model itself (without the SIHFR defense method).

The average processing time (per image), at a 95% confidence interval, taken by the SIHFR method was 1.65 ± 0.22 ms whereas that consumed by the ResNet50-based VMMR model (without SIHFR) was around 5.33 ± 0.08 ms. Activating the SIHFR module slightly increased the average VMMR processing time to 6.98 ± 0.30 ms, adding an overhead of 27.33% – 34.62% which amounts to less than 2ms per image (on average). It is also worth mentioning that SIHFR does not require re-training of the CNN-based VMMR model and can work as a connected complementary module, potentially serving as a lightweight trigger to launch more advanced or sophisticated defenses.

6.4.3 Comparative Study

This paper focuses on developing a lightweight defense method that pre-processes input images to eliminate or reduce the influence of adversarial patches. In doing so, two main design goals that need to be satisfied are: (i) the defense method should not require re-training or modifications of the CNN-based VMMR model, and (ii) the processing time required by the defense method, per image, should be minimal. From the few defense methods that have been proposed in the literature to mitigate

Table 17 Evaluating SIHFR against P4C- and P5A-based attacks

Patch/region	Avg. recall (no SIHFR)	Avg. recall (with SIHFR)	Avg. improvement (%)
No Patch Attack	0.9862 ± 0.01	0.9657 ± 0.01	-2.1
P4C-R5	0.4941 ± 0.05	0.8365 ± 0.01	69.28
P5A-R5	0.6719 ± 0.04	0.8080 ± 0.02	20.25

the effect of adversarial patches through input image pre-processing, the recent work of [1] is closest in spirit to our work.

The major factor we examine when comparing SIHFR with Ally-Patches of [1] is the average processing time consumed by the methods per input image. Using a common computing platform, without any hardware accelerators such as GPUs, we run both methods on test images attacked with adversarial patches. While the SIHFR method designed and developed in this paper consumed on average, $2.60 \pm 0.13ms$ (at a 95% confidence interval), the Ally-Patches method required $922.26 \pm 32.78ms$ on average. The slow performance of Ally-Patches is due to the sliding window-based approach to extract candidate ally patches followed by a procedure to filter out some patches which is based on overlap criteria or mutual information [35] constraints.

It is worth mentioning that the method proposed in [1], unlike SIHFR, requires the CNN-based detection models to be trained to detect or classify objects based on incomplete and partial views (patches) of objects of interest. However, this requirement puts an additional limitation for CNN-based VMMR models due to the multiplicity and ambiguity challenges.

6.4.4 Effect of Patch Size on Defense Robustness

In these set of experiments, we investigate the robustness of SIHFR defense method against adversarial patches placed at five different scales with the R5 region on the image. The five scales for the patches used in these experiments are: 25×25 , 40×40 , 50×50 , 60×60 , and 70×70 . We choose to conduct the evaluations with the two best patches developed: P4C and P5A.

In Fig. 12, we provide the results in terms of average precision, recall and F1 scores, with and without the SIHFR defense method against attacks launched using different sizes of P4C and P5A. As one may observe, with adversarial patches of sizes above 40×40 , SIHFR improves the performance scores of VMMR. This demonstrates the effectiveness of SIHFR in improving the robustness of the CNN-based VMMR model. Moreover, we find that the SIHFR method is highly robust against adversarial patches of sizes as low as 50×50 and as large as 70×70 , given input images of sizes 224×224 .

With small adversarial patches, e.g. of size 25×25 , though their impact on VMMR performance is limited, the benefits of activating SIHFR defense method are not that evident. However, looking at the attack detection rates of SIHFR (in Fig. 13), we note that SIHFR still succeeded in detecting around 72% of attacks launched using P4C and around 68% of attacks launched using P5A, at a patch size of 25×25 .

7 Conclusion and Future Work

In this paper, we studied and introduced a cyber-physical attack based on adversarial patches that could be placed on vehicles to fool or circumvent an important component of a safe and secure smart city infrastructure, i.e., an automated vehicle

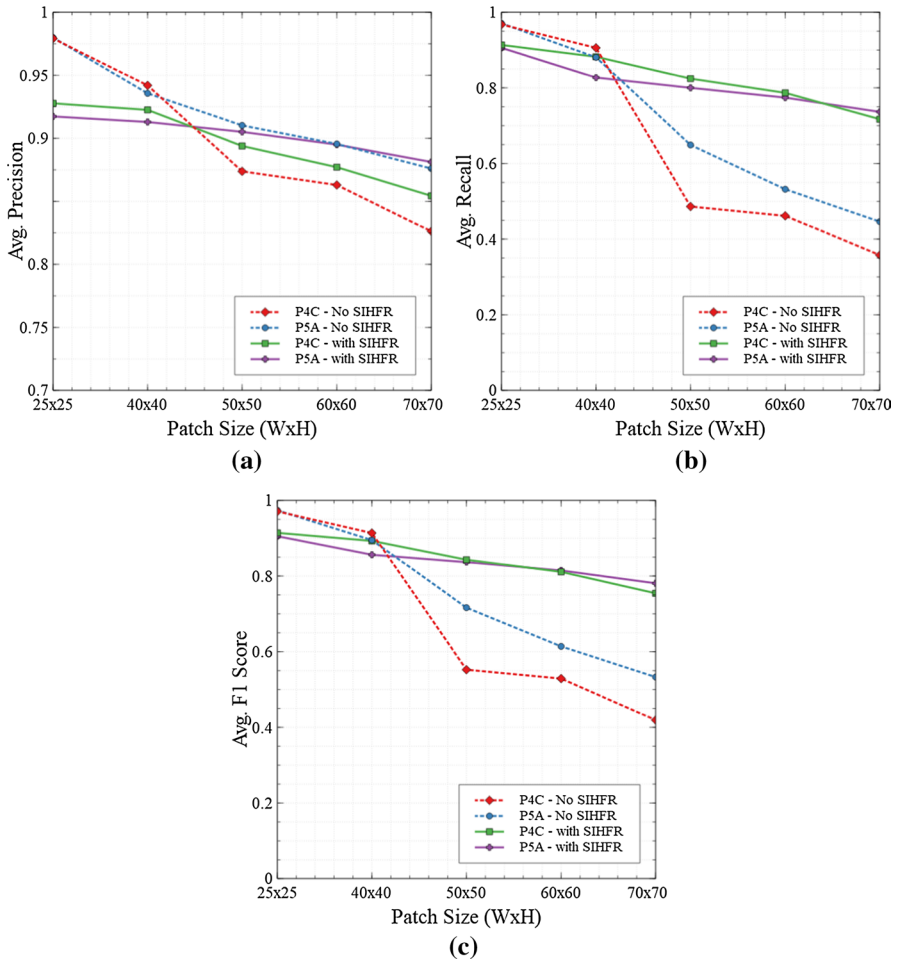


Fig. 12 (Color online) Effect of patch size on SIHFR's robustness, measured in terms of improvements in VMMR performance: **a** average Precision, **b** average recall, and **c** average F1 scores. The attacks were launched using P4C and P5A. Higher the score, in comparison to the case with no SIHFR defense, the higher is SIHFR's robustness

surveillance system such as VMMR. Through experimental evaluations on a real-world surveillance nature dataset of vehicles, we demonstrated the effectiveness of the developed adversarial patches in reducing the classification performance of a robust VMMR system that is based on a popular convolutional neural network model known as ResNet50. A reduction of up to 48% in recall score was achieved by the developed patches. In addition, we evaluated two patch placement strategies: fixed (at top-left) vs. random, and found that adversarial patches placed at random locations in the image during training and execution are more effective. Moreover, we designed and developed a lightweight defense method called *SIHFR* to eliminate the effect of adversarial patches-based attacks on VMMR systems by making use of

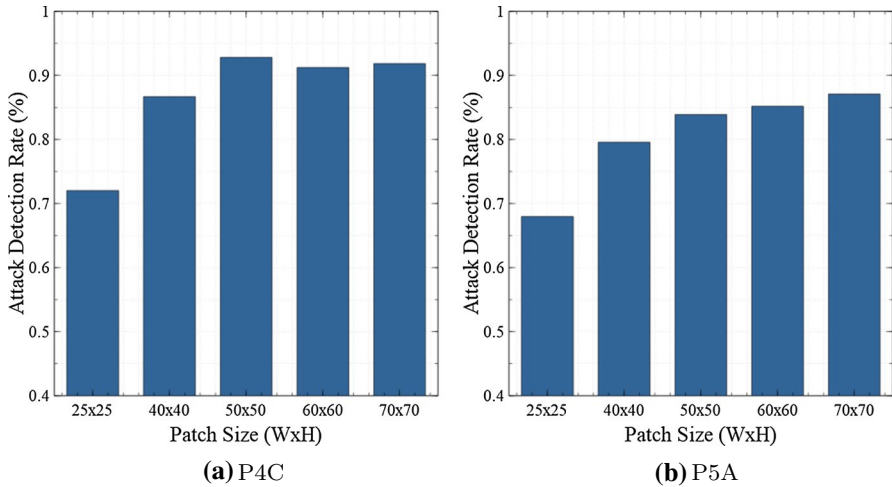


Fig. 13 Comparing SIHFR's attack detection rate at different sizes of adversarial patches P4C and P5A

symmetry in vehicle images. The proposed defense method succeeded in defending against 83.40% to 92.15% of the attacks (with patches of size 50×50), leading to an improvement in VMMR performance by up to 69.28%.

To the best of our knowledge, this is the first work that investigates the problem of adversarial learning against VMMR systems and proposes a light-weight defense method for the same. It is hoped that this work shall pave the path forward for future studies in developing VMMR systems that are highly robust to cyber-physical attacks based on adversarially learnt patches, in a quest to achieve more secure and adversarially robust smart city surveillance systems. In future, we plan to investigate multi-adversarial patches-based attacks that target VMMR systems.

Acknowledgement This study was partially funded by Canada Research Chairs Program and Natural Sciences and Engineering Research Council of Canada (NSERC)'s CREATE TRANSIT Program.

References

1. Abdel-Hakim, A.E.: Ally patches for spoilation of adversarial patches. *J. Big Data* **6**, 51 (2019)
2. Thys, S., Ranst, W.V., Goedemé, T.: Fooling automated surveillance cameras: Adversarial patches to attack person detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 49–55 (2019)
3. Huang, L., Gao, C., Zhou, Y., Xie, C., Yuille, A.L., Zou, C., Liu, N.: Universal physical camouflage attacks on object detectors. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, pp. 717–726. IEEE (2020)
4. Duan, R., Ma, X., Wang, Y., Bailey, J., Qin, A.K., Yang, Y.: Adversarial camouflage: Hiding physical-world attacks with natural styles. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, pp. 997–1005. IEEE (2020)
5. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramèr, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: 12th USENIX Workshop on

- Offensive Technologies, WOOT 2018, Baltimore, MD, USA, August 13–14, 2018. USENIX Association (2018)
6. Boukerche, A., Siddiqui, A.J., Mammeri, A.: Automated vehicle detection and classification: models, methods, and techniques. *ACM Comput. Surv.* **50**(5) (2017)
 7. Boukerche, A., Hou, Z.: Object detection using deep learning methods in traffic scenarios. *ACM Comput. Surv.* **54**(2), 1–35 (2021)
 8. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540. Association for Computing Machinery, New York, NY, USA (2016)
 9. Brown, T., Mane, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. In: *Conference on Neural Information Processing Systems (NuerIPS), Machine Learning and Computer Security Workshop (Poster)* (2017)
 10. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525 (2017)
 11. Liu, X., Yang, H., Liu, Z., Song, L., Chen, Y., Li, H.: DPATCH: an adversarial patch attack on object detectors. *CEUR-WS.org* (2019)
 12. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
 13. Naseer, M., Khan, S., Porikli, F.: Local gradients smoothing: Defense against localized adversarial attacks. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1300–1307. IEEE Computer Society, Los Alamitos, CA, USA (2019)
 14. Hayes, J.: On visible adversarial perturbations digital watermarking. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1678–16787 (2018)
 15. Das, N., Shanbhogue, M., Chen, S.T., Hohman, F., Li, S., Chen, L., Kounavis, M.E., Chau, D.H.: Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In: *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–204. ACM, New York, NY, USA (2018)
 16. Guo, C., Rana, M., Cissé, M., van der Maaten, L.: Countering adversarial images using input transformations. In: *6th International Conference on Learning Representations, ICLR (Poster)* (2018)
 17. Aloqaily, M., Otoum, S., Ridhawi, I.A., Jararweh, Y.: An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks* **90**, 101842 (2019). Recent advances on security and privacy in Intelligent Transportation Systems
 18. Kalbo, N., Mirsky, Y., Shabtai, A., Elovici, Y.: The security of ip-based video surveillance systems. *Sensors* **20**(17), 4806 (2020)
 19. Kumar, A.R., Sivagami, A.: Security aware multipath routing protocol for wmsns for minimizing effect of compromising attacks. *J. Netw. Syst. Manag.* **27**(3), 573–599 (2019)
 20. Lahrouni, Y., Pereira, C., Bensaber, B.A., Biskri, I.: Using mathematical methods against denial of service (dos) attacks in VANET. In: *15th ACM International Symposium on Mobility Management and Wireless Access, MOBIWAC 2017*, pp. 17–22. ACM (2017)
 21. Salameh, H.B., Derbas, R., Aloqaily, M., Boukerche, A.: Secure routing in multi-hop iot-based cognitive radio networks under jamming attacks. In: *22nd Int'l ACM Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 323–327. ACM (2019)
 22. Siddiqui, A.J., Boukerche, A.: Adaptive ensembles of autoencoders for unsupervised iot network intrusion detection. *Computing* (2021)
 23. Li, J., Liang, W., Xu, W., Xu, Z., Zhao, J.: Maximizing the quality of user experience of using services in edge computing for delay-sensitive iot applications. In: *23rd Int'l ACM Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 113–121. ACM (2020)
 24. Thomas, D., Shankaran, R.: A secure barrier coverage scheduling framework for wsn-based iot applications. In: *23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 215–224. ACM (2020)
 25. Boukerche, A., Machado, R.B., Jucá, K.R.L., Sobral, J.B.M., Notare, M.S.M.A.: An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Comput. Commun.* **30**(13), 2649–2660 (2007)
 26. Boukerche, A., Jucá, K.R.L., Sobral, J.B.M., Notare, M.S.M.A.: An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Comput.* **30**(5–6), 629–646 (2004)

27. Boukerche, A., Notare, M.S.M.A.: Behavior-based intrusion detection in mobile phone systems. *J. Parallel Distrib. Comput.* **62**(9), 1476–1490 (2002)
28. Tan, L., Xiao, H., Yu, K., Aloqaily, M., Jararweh, Y.: A blockchain-empowered crowdsourcing system for 5g-enabled smart cities. *Comput. Stand. Interfaces* **76**, 103517 (2021)
29. Chen, Q., Srivastava, G., Parizi, R.M., Aloqaily, M., Ridhawi, I.A.: An incentive-aware blockchain-based solution for internet of fake media things. *Inf. Process. Manag.* **57**(6), 102370 (2020)
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp. 770–778. IEEE Computer Society (2016)
31. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5188–5196 (2015)
32. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
33. Siddiqui, A.J., Mammeri, A., Boukerche, A.: Real-time vehicle make and model recognition based on a bag of surf features. *Trans. Intell. Transport. Syst.* **17**(11), 3205–3219 (2016)
34. Yang, L., Luo, P., Loy, C.C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3973–3981 (2015)
35. Russakoff, D.B., Tomasi, C., Rohlffing, T., Jr., C.R.M.: Image similarity using mutual information of regions. pp. 596–607. Springer (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Abdul Jabbar Siddiqui¹  · Azzedine Boukerche¹

Azzedine Boukerche
boukerch@site.uottawa.ca

¹ PARADISE Lab, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, Canada